

# Documentación Practico 1



## Programación 2 Grupo A

### EQUIPO SINGAPUR



Integrantes del equipo:

Juan Pablo Bella

Ignacio Dodera

Guillermo Mercadal

Gustavo Mayer

Francisco Rostan

Agustin Vera

## EJERCICIO 1:

Se plantea un programa para la creación de un vehículo, en el cual se podrá modificar su tapicería y tipo de climatizador.

Este programa presenta 2 opciones, las cuales son;

1. Crear un vehículo; Esta opción va mostrando opciones al usuario para que ingrese las características del vehículo a crear mediante el ingreso de números para seleccionar la opción de preferencia para el usuario.

Estas características serán:

- Color de la carrocería (1. Rojo. 2. Gris. 3. Blanco. 4. Azul.)
- Color del interior (1. Rojo. 2. Gris. 3. Blanco. 4. Camel.)
- Tipo de climatizador (1. Caliente. 2. Frio.)

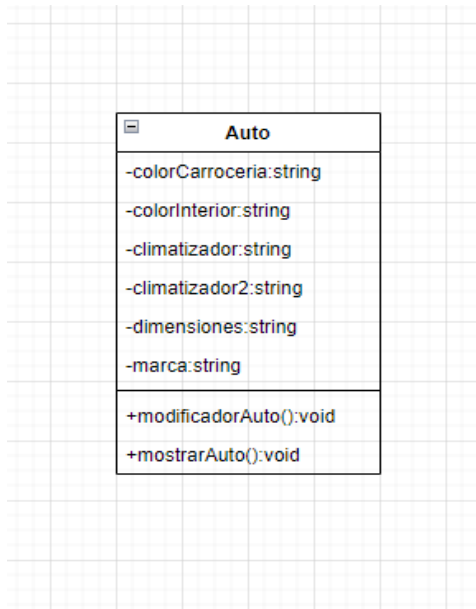
También se le pedirá al usuario que ingrese los datos de;

- Medidas del vehículo (Largo y ancho)
- Marca del auto.

2. Mostrar vehículo; Una vez ingresados los datos, el programa mostrara la información ingresada en cada campo de la opción 1. También esta opción dará fin al programa al presionar cualquier tecla.

Se utilizó una clase llamada Auto para almacenar los datos que elegirá el usuario sobre el vehículo a crear. La misma contiene un método que define las características que ingreso el usuario.

También se creó un método el cual se encarga de mostrar las características finales del vehículo.



## EJERCICIO 2:

Se crea un programa en el cual se pueda ingresar el nombre y edad de dos alumnos de una institución, y que los mismos se impriman en pantalla si es pedido por el usuario.

Este programa consiste en el uso de un menú que te da la opción de;

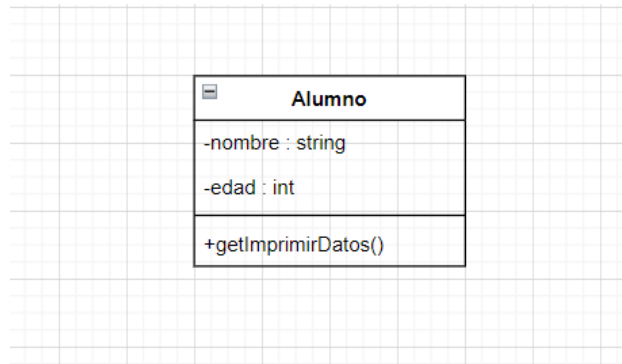
1. Ingresar nombre; Guarda el nombre del primer y segundo alumno (siempre pide los dos obligatoriamente), si ya tiene ingresados los nombres, la opción sustituye los anteriores nombres, por los nuevos ingresados por el usuario.
2. Ingresar edad; Guarda la edad de los alumnos de la misma manera que con los nombres (de la misma manera, si el usuario ya ingreso las edades anteriormente, el programa sustituye las anteriores edades con las nuevas ingresadas).
3. Mostrar información; Muestra los datos que han sido ingresados por el usuario hasta ese momento en el programa. Si el usuario ha ingresado tanto nombre como edad de los alumnos, el mismo imprime ambos datos, si solo se ingresan los nombres el programa muestra un 0 en el campo de edades.

De manera similar, si solo se ingresan las edades, el campo de nombre se muestra vacío.

4. Terminar bucle; Se cierra el programa.

Se utilizó una clase llamada Alumno para definir el nombre y la edad del alumno. Esta misma clase también contiene un método el cual realiza la impresión de los dos datos en conjunto.

Se consideró utilizar un menú para mejorar la usabilidad para el usuario.



## EJERCICIO 3 y 4:

Se plantea crear un programa el cual puede realizar la suma o resta de dos vectores.

El mismo va a consistir de un menú el cual tendrá 4 opciones que puede elegir el usuario.

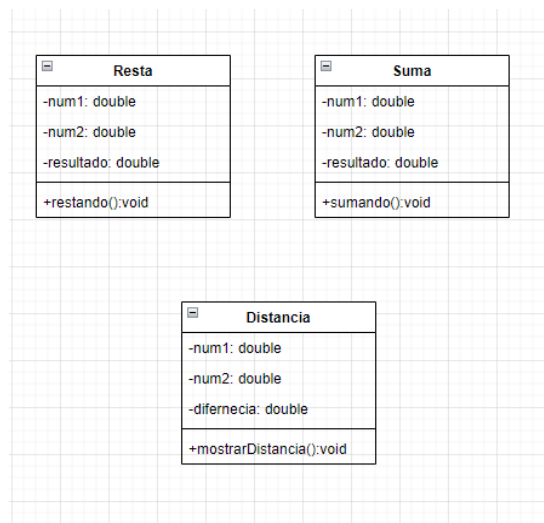
1. Sumar: Realiza la suma de dos vectores los cuales serán pedidos al usuario.
2. Restar: Realiza la resta de dos vectores que, de igual manera, serán pedidos al usuario.
3. Calcular distancia: Realiza un cálculo de la distancia numérica entre los dos vectores que ingrese el usuario
4. Terminar: Cierra el programa

Para realizar el programa utilizamos 3 clases:

- La clase Suma; la cual guarda los 2 números ingresados por el usuario y mediante un método devuelve el resultado de la suma de los números.

- La clase Resta; la cual tiene el mismo funcionamiento que la clase Suma, pero el método dentro de ella devuelve la resta entre los números ingresados.
- La clase Distancia; que funciona similar a las 2 clases anteriores, pero el método que funciona dentro de ella contiene la lógica para calcular la distancia ya sea positiva o negativa entre ambos números.

Se decidió plantear la suma y resta de vectores, en conjunto con el cálculo de la distancia dentro del mismo menú. Esto con la intención de facilitar la usabilidad para el usuario.



## EJERCICIO 5:

Se crea un programa el cual tiene la funcionalidad de sumar dos matrices mediante el ingreso de datos por parte del usuario.

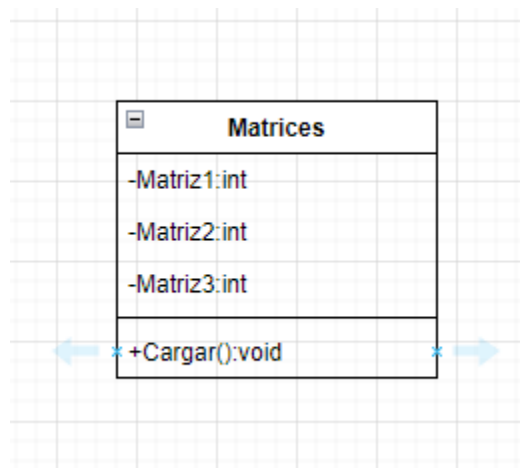
El programa empieza pidiendo la cantidad de números enteros que va a contener las filas y las columnas de las matrices (ambas matrices son del mismo número de filas y columnas que define el usuario al principio).

Posteriormente pide el valor de cada posición de ambas matrices (especificando cual se está pidiendo en el momento del ingreso).

Una vez completa la matriz uno y la matriz dos, se suman y se muestra una matriz con el resultado de dicha suma.

Para realizar el programa utilizamos una única clase Matrices que contiene la lógica de todo el programa.

A su vez se empleó un solo método que toma los ingresos de datos del usuario, crea y suma las matrices (si bien esto corresponde más al curso anterior se podría emplear objetos en una posterior mejora).



## EJERCICIO 6:

Se plantea la creación de un programa el cual pueda almacenar la hora con precisión de milisegundos.

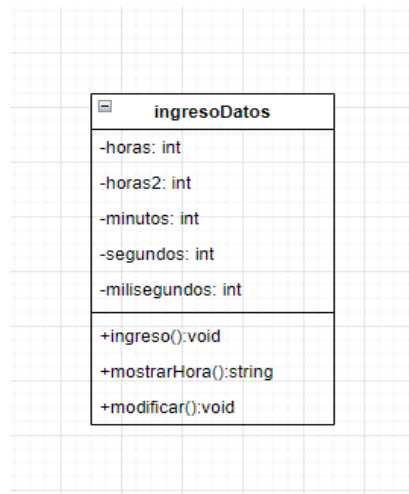
El programa consiste en un menú que mediante el ingreso de números permite al usuario elegir si desea ingresar los datos, mostrar los datos ya ingresados o modificarlos, como también dar finalización al programa.

1. Para introducir datos; El usuario puede ingresar los datos de la hora que se quiere imprimir (hora, minutos, segundos, milisegundos).
2. Para mostrar el horario; Imprime los datos guardados de cada campo hasta el momento.

3. Para modificarlos; Esto permite modificar totalmente cada campo de la hora, o en su defecto hacer una suma a un campo anteriormente ingresado.
4. Terminar el programa.

Se utilizó una clase para el ingreso de datos, la cual almacena la hora que quiere imprimir el usuario. Esta misma contiene 3 métodos los cuales se encargan del ingreso de datos, de mostrar la hora y de la modificación de datos.

Además, se creó un método que se encarga de la verificación de los datos ingresados para que estén dentro de los valores comprendidos de cada campo de la hora.



## EJERCICIO 7:

Se crea un programa que permite el ingreso de distinto tipo de documentación (remitos, facturas, recibos de sueldos e impuestos). Este programa tiene la utilidad de servir como repositorio para agrupar todos estos documentos digitalmente y poder imprimirlos cuando se necesiten.

El programa consiste en un menú que permite al usuario elegir entre el ingreso de datos o mostrar los datos ya ingresados.

Para realizar el programa utilizamos 5 clases:

- La clase Nota de entrega la cual permite el ingreso de la cantidad de bultos, el número de la entrega y la fecha.
- La clase Factura que permite ingresar la fecha y el importe de las facturas.
- La clase Factura Luz que permite ingresar el código del pago y el importe de la factura de la luz de la empresa.
- La clase Municipal sirve para ingresar el importe y la fecha de partida del importe municipal.
- La clase Recibo Sueldo permite el ingreso de los legajos y el monto total de los recibos de sueldos.

Todas las clases están compuestas por 2 métodos, el primero para el ingreso de datos y el segundo para devolver los datos ingresados.

